PATENT COOPERATION TREATY

From the INTERNATIONAL BUREAU

PCT

NOTIFICATION OF ELECTION

(PCT Rule 61.2)

To:

Assistant Commissioner for Patents
United States Patent and Trademark Office
Box PCT
Washington, D.C.20231
ETATS-UNIS D'AMERIQUE

in its capacity as elected Office

Date of mailing: 05 October 2000 (05.10.00)	
International application No.: PCT/GB00/00882	Applicant's or agent's file reference: A25721 WO
International filing date: 10 March 2000 (10.03.00)	Priority date: 31 March 1999 (31.03.99)
Applicant: WARD, Richard, Beresford et al	

1. The designated Office is hereby notified of its election made:

[X] in the demand filed with the International preliminary Examining Authority on:

14 August 2000 (14.08.00)

[ ] in a notice effecting later election filed with the International Bureau on:

2. The election [X] was

[ ] was not

made before the expiration of 19 months from the priority date or, where Rule 32 applies, within the time limit under Rule 32.2(b).

The International Bureau of WIPO
34, chemin des Colombettes
1211 Geneva 20, Switzerland

Facsimile No.: (41-22) 740.14.35

Authorized officer:

J. Zahra

Telephone No.: (41-22) 338.83.38

PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 7 : G06F 11/36 — A1

(11) International Publication Number: WO 00/58836

(43) International Publication Date: 5 October 2000 (05.10.00)

(21) International Application Number: PCT/GB00/00882

(22) International Filing Date: 10 March 2000 (10.03.00)

(30) Priority Data:

9907438.7	31 March 1999 (31.03.99)	GB
99305543.3	13 July 1999 (13.07.99)	EP

(71) Applicant *(for all designated States except US)*: BRITISH TELECOMMUNICATIONS PUBLIC LIMITED COMPANY [GB/GB]; 81 Newgate Street, London EC1A 7AJ (GB).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: WARD, Richard, Beresford [GB/GB]; Tangley Way, The Street, Witnesham, Ipswich, Suffolk IP6 9HG (GB). GRAHAM, John, Alexander [GB/GB]; 41 Coronation Road, Ipswich, Suffolk IP4 5EN (GB). AYLETT, Martin, Richard [GB/GB]; Dale View, Mendham, Harleston, Norfolk IP20 0JN (GB).

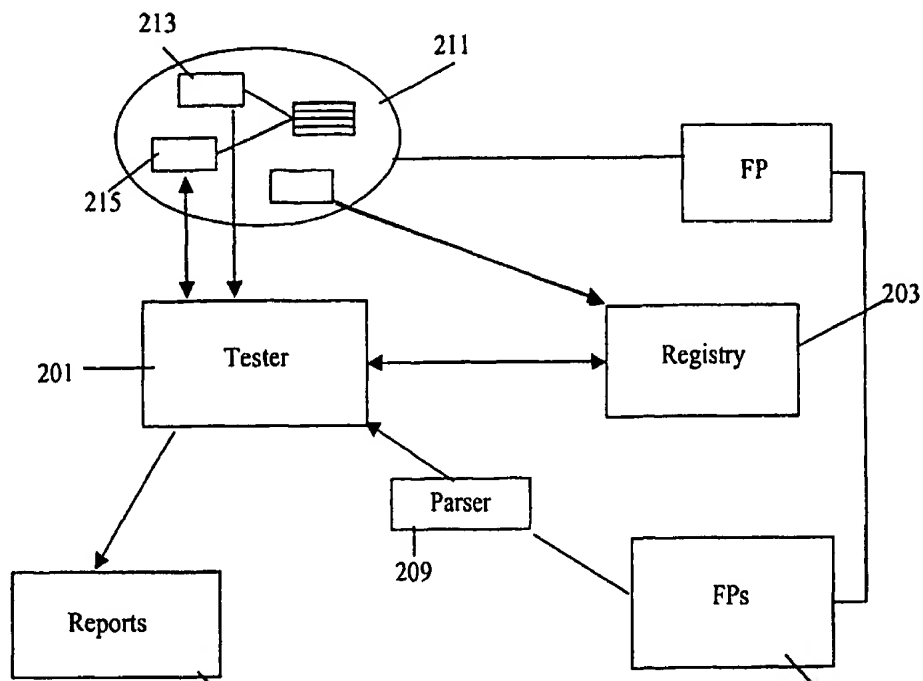
(74) Agent: ROBINSON, Simon, Benjamin; BT Group Legal Services, Intellectual Property Department, Holborn Centre, 8th floor, 120 Holborn, London EC1N 2TE (GB).

(81) Designated States: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published

With international search report.

(54) Title: METHOD AND APPARATUS FOR AUTOMATED SOFTWARE TESTING



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	**ES**	Spain	**LS**	Lesotho	**SI**	Slovenia
AM	Armenia	**FI**	Finland	**LT**	Lithuania	**SK**	Slovakia
AT	Austria	**FR**	France	**LU**	Luxembourg	**SN**	Senegal
AU	Australia	**GA**	Gabon	**LV**	Latvia	**SZ**	Swaziland
AZ	Azerbaijan	**GB**	United Kingdom	**MC**	Monaco	**TD**	Chad
BA	Bosnia and Herzegovina	**GE**	Georgia	**MD**	Republic of Moldova	**TG**	Togo
BB	Barbados	**GH**	Ghana	**MG**	Madagascar	**TJ**	Tajikistan
BE	Belgium	**GN**	Guinea	**MK**	The former Yugoslav Republic of Macedonia	**TM**	Turkmenistan
BF	Burkina Faso	**GR**	Greece			**TR**	Turkey
BG	Bulgaria	**HU**	Hungary	**ML**	Mali	**TT**	Trinidad and Tobago
BJ	Benin	**IE**	Ireland	**MN**	Mongolia	**UA**	Ukraine
BR	Brazil	**IL**	Israel	**MR**	Mauritania	**UG**	Uganda
BY	Belarus	**IS**	Iceland	**MW**	Malawi	**US**	United States of America
CA	Canada	**IT**	Italy	**MX**	Mexico	**UZ**	Uzbekistan
CF	Central African Republic	**JP**	Japan	**NE**	Niger	**VN**	Viet Nam
CG	Congo	**KE**	Kenya	**NL**	Netherlands	**YU**	Yugoslavia
CH	Switzerland	**KG**	Kyrgyzstan	**NO**	Norway	**ZW**	Zimbabwe
CI	Côte d'Ivoire	**KP**	Democratic People's Republic of Korea	**NZ**	New Zealand		
CM	Cameroon			**PL**	Poland		
CN	China	**KR**	Republic of Korea	**PT**	Portugal		
CU	Cuba	**KZ**	Kazakstan	**RO**	Romania		
CZ	Czech Republic	**LC**	Saint Lucia	**RU**	Russian Federation		
DE	Germany	**LI**	Liechtenstein	**SD**	Sudan		
DK	Denmark	**LK**	Sri Lanka	**SE**	Sweden		
EE	Estonia	**LR**	Liberia	**SG**	Singapore		

Method and Apparatus for Automated Software Testing

The present invention relates to software testing and in particular to unit testing software during its operation. The invention can be applied advantageously, but not exclusively, to software produced using object oriented programming languages such as C++, Corba or Java.

Automated testing of software during its development is known. The tests are designed as part of a software development process and these are then programmed into specialised test tools and executed automatically. Many tools are commercially available to support this type of software development technique.

Software that checks itself during operation is also known and has been developed and applied widely. This may involve checking pre and post-conditions or assertions and looking for exceptions at appropriate points in the software during its normal execution (See "Self Testing Systems" - M Aylett and P Utton, BT Technology Journal 1992).

Known testing systems enable end-to-end tests to be run on operational software systems in order to test out the operation of individual facilities. However, there are currently no testing systems that easily enable low level tests to be run on a fully integrated and operational system. These tests are often termed "unit tests" and are applied directly to one or more individual units of code (e.g. a function, method, module or agent). This is in contrast to end-to-end tests of a system that run from a system or user interface. Unit tests are currently run manually or automatically during development before integration.

According to the present invention there is provided a method of testing an operational integrated software system, said system comprising a plurality of software elements, said method comprising the steps of:

b) associating a set of test criteria with each registered element of software;

c) selecting an element registered in the registry and testing the element in accordance with the associated set of test criteria; and

d) capturing the results of the testing of the element and comparing them to the associated test criteria.

This provides the advantage of enabling unit testing to be carried out on an integrated software system during its operation that allows quick identification of latent or newly introduced faults in the software.

Figure 1 is a schematic representation of a computer loaded with software embodying the present invention;

Figure 2 is a functional block diagram of the program elements that comprise the software indicated in Figure 1;

Figure 3 is a flow diagram illustrating part of the processing of the software shown in Figure 2;

Figures 4a and 4b are tables illustrating the data structures used and created by the program elements shown in Figure 2; and

Figure 5 is a flow diagram showing a further part of the processing of the software shown in Figure 2.

Figure 1 illustrates a conventional computer 101 such as a PC, running a conventional operating system 103 such as Windows and having a number of resident application programs 105 such as a word processing program, a network browser and e-mail program or a database management program. The computer 101 also includes a software development application program 107 that enables the user to write and compile new programs and a testing program 109 that enables testing to be carried out on programs. The computer 101 is also connected to a conventional disc storage unit 111 for storing data and programs, a keyboard 113 and mouse115 for allowing user input and a printer 117 and display unit 119 for providing output from the computer 101. The computer 101 also has access to external networks (not shown) via a network card 121.

In conventional object oriented programming the programs are divided into conceptual sub-units called objects. Each object carries out predetermined functions much in the same way that a sub-routine might in conventional programming. Objects carry out processing of data and may co-operate with other objects to carry out some functions. Such co-operation is carried out via interfaces between the objects called arguments that are provided for passing commands, requests and data between th objects.

Each object is categorised into a class of objects. In fact, it is the class of an object that determines the functions and performance of an object. An object itself is an embodiment (or instance) of the class and can be created to carry out its function and then deleted once the function is complete. The creation of an object for a given class is carried out under the control of a constructor algorithm. In addition, the corresponding destructor for each class is arranged to remove the entry when th corresponding object is deleted.

Each object comprises one or more methods. Each method is a subroutine that together with other methods provides the functions of the object itself. Methods may co-operate with other objects to carry out functions/processing on behalf of the method. The methods are also defined by the class of the object as are the arguments of the object.

In summary, objects are functional units of software code whose functions are defined by the class of which a given object is an instance. Objects can have a number of states that change depending on the object's interaction with other objects or data. The combined interaction of the objects that make up a computer program provide the functions of the program itself.

With reference to figure 2, the testing program 109 comprises five main components, a tester 201, an object registry 203, a report generator 205, a test criteria store 207 and a parser 209. The tester 201 carries out the testing of each object in the

generator 205. The object registry 203 provides the tester 201 with a list of the objects that form part of the program at any given time (as noted above, objects may be created and destroyed during the operation of a program). The test criteria store 207 is used to hold the data and/or instructions necessary to test each of the objects registered in the object registry 203. In the present embodiment the data and/or instructions held in the test criteria store 207 are immediately usable by the tester 201. However, in some cases the data may be coded using a scripting language. In this case the parser 209 would be used to convert the data/instruction into a form usable by the tester 201. The functions and interactions of the five main components will be described in further detail below.

Figure 2 also shows a program object 211 undergoing testing by the tester 201. The object 211 is a standard object but has three additional areas of functionality that allow it to interact automatically with the testing program 109. The added functionality is provided in the present embodiment by two special methods 213, 215 added to each class definition used in the program under test and by additions to the functionality of the constructor and destructor algorithms for the program.

With reference to figure 3, the constructor is arranged, on the instantiation of an object for a given class, to create an entry in the object registry 203 for the new object (see step 301 of chart C). Then, at step 303, the constructor enters the identification for the object in its entry in the registry 203 (each object, when it is constructed by the constructor, is assigned a unique identifier). At step 305, the class type of the object is entered in the entry for the object and at step 307 the corresponding class name is entered. After step 307 the registration process is completed and the constructor algorithm ends its processing.

As noted above, when an object is no longer required it is deleted by a destructor algorithm. In the present embodiment, the destructor algorithm is also arranged to carry out the steps shown in chart D of Figure 3. At step 309 the destructor algorithm identifies the entry in the registry 203 that corresponds to the object being deleted and at step 311 removes the entry from the registry 203.

With reference to figure 4a, each class of object has a test criteria file that is entered into the test criteria store 207 when the first object of that class is entered in the object registry 203. The criteria are created during the design and implementation of the computer program under test and their precise construction is dependent on the testing methods being used. In the present embodiment, an entry is made in the store 207 for each class 401. For every class, an entry 403 is made for each method within the class. For each method 403, a definition of the input 405 to the method, the output 407 from the method, the start state 409 of the object when the method is performed and the end state 411 of the object on completion of the method is entered in the store 207.

The operation of the tester 201 will be described now with reference to Figure 5 in which at step 501 the tester 201 awaits a command to commence testing. In the present embodiment the command is given by a user. Once the command has been received then, at step 503, the tester 201 chooses the class of object to be tested from the registry 203. In the present embodiment, the system responds to a user command to commence testing and the chooses a method at random. However, the command or choice of method could be produced randomly, in accordance with a predefined testing plan or in response to requests or events from other objects or programs.

At step 505 the tester 201 uses the first special method 213 to determine the number of methods in the chosen object. The method 213 returns data, as shown in Figure 4b, describing the class of the object 413, identifications 415 of each of the methods in the object and a description 417 of the arguments for each of the method. At step 507, using the class identification returned by the method 213, the tester 201 identifies the appropriate test criteria from the test criteria store 207 and at step 509 runs the chosen method against the identified test criteria.

At step 511, the tester 201 uses the second special method 215 to capture the results of the test run on the method. The precise data that is captured is determined by the test criteria and may include the output data from the tested method, the

methods that the chosen method interacted with as a result of the test. At step 513, the test data collected in the previous step is compared to the test criteria and the results of the comparison are passed to the report generator 205 for inclusion in a test report. After step 513, the tester returns to step 501 to await a further test instruction.

The tester program 201 is designed to carry out its testing procedures on a program while the program is in operation. In some operating systems the testing program 201 could be arranged to run as a background process or be arranged to operate when there is a predetermined amount of spare processor resource available.

As will be understood by those skilled in the art, in some systems it may be necessary to include means for preventing changes to the run-time environment being made during the testing of a software element. These may be in the form of run-time test switches that are similar in function to a debug compiler switch. In some systems it may be necessary to include a means to restore the state of any persistent variables (variables that retain state after execution) affected by the tests. This can be performed by taking a copy of the persistent variables before a test and restoring them afterwards.

It will also be clear to those skilled in the art that the system under test could distributed in nature. For example, testing could be carried out over a network and units of code distributed across many computers. Also, the testing system can be used by developers during the design and build of a software system or be provided as part of the functionality of programs that are ready for use.

The tester program is preferably written in the same language as the program that it is testing. However, although the embodiment above describes the testing of an object oriented programming language, it will be understood by those skilled in the art that the principles of the invention are also applicable to other programming languages. Other such languages may be modular programming languages (such as Modula-2) or sequential programming languages (such as Pascal). It also [illegible]

understood that the term "object" used in the this description is to be construed broadly so as to cover functions, methods, modules or agents.

As will be understood by those skilled in the art, the tester program 109 can be contained on various transmission and/or storage mediums such as a floppy disc, CD-ROM, or magnetic tape so that the program can be loaded onto one or more general purpose computers or could be downloaded over a computer network using a suitable transmission medium.

Unless the context clearly requires otherwise, throughout the description and the claims, the words "comprise", "comprising" and the like are to be construed in an inclusive as opposed to an exclusive or exhaustive sense; that is to say, in the sense

CLAIMS

1. A method of testing an operational integrated software system, said system comprising a plurality of software elements, said method comprising the steps of:

a) automatically registering each active element of software in a registry;

b) associating a set of test criteria with each registered element of software;

c) selecting an element registered in the registry and testing the element in accordance with the associated set of test criteria; and

d) capturing the results of the testing of the element and comparing them to the associated test criteria.

2. A method according to Claim 1 in which each element of software is arranged to automatically register an identification of itself in the registry.

3. A method according to Claims 1 or 2 in which each element of software is arranged to capture the results of its testing.

4. A method according to any of Claims 1 to 3 further comprising the step of automatically providing a report on the results of the testing.

5. A method according to any preceding claim in which the test criteria are defined using a scripting language and said method further comprises the step of parsing the test criteria to convert them into a form for testing against.

6. An apparatus for testing an operational integrated software system, said system comprising a plurality of software elements, said apparatus comprising:

a) means for the automatic registration of each active element of software;

c) means for selecting a registered element of software and testing the element in accordance with the associated test criteria; and

d) means for comparing the results of the testing of the element against the associated test criteria.

7. An apparatus according to Claim 6 in which each element of software is provided with means for automatically registering itself.

8. A method according to Claim 6 or 7 in which each element of software is provided with means for capturing the results of its testing.

9. An apparatus according to any of Claims 6 to 8 further comprising means for producing a report of the results of testing an element of software.

10. An apparatus according to any of Claims 6 to 9 in which the test criteria are defined using a scripting language and the apparatus further comprises means for parsing the test criteria to convert them into a form for testing against.

11. A data carrier loadable into a computer and carrying instructions for causing the computer to carry out the method according to Claim 1.

12. A data carrier loadable into a computer and carrying instructions for enabling

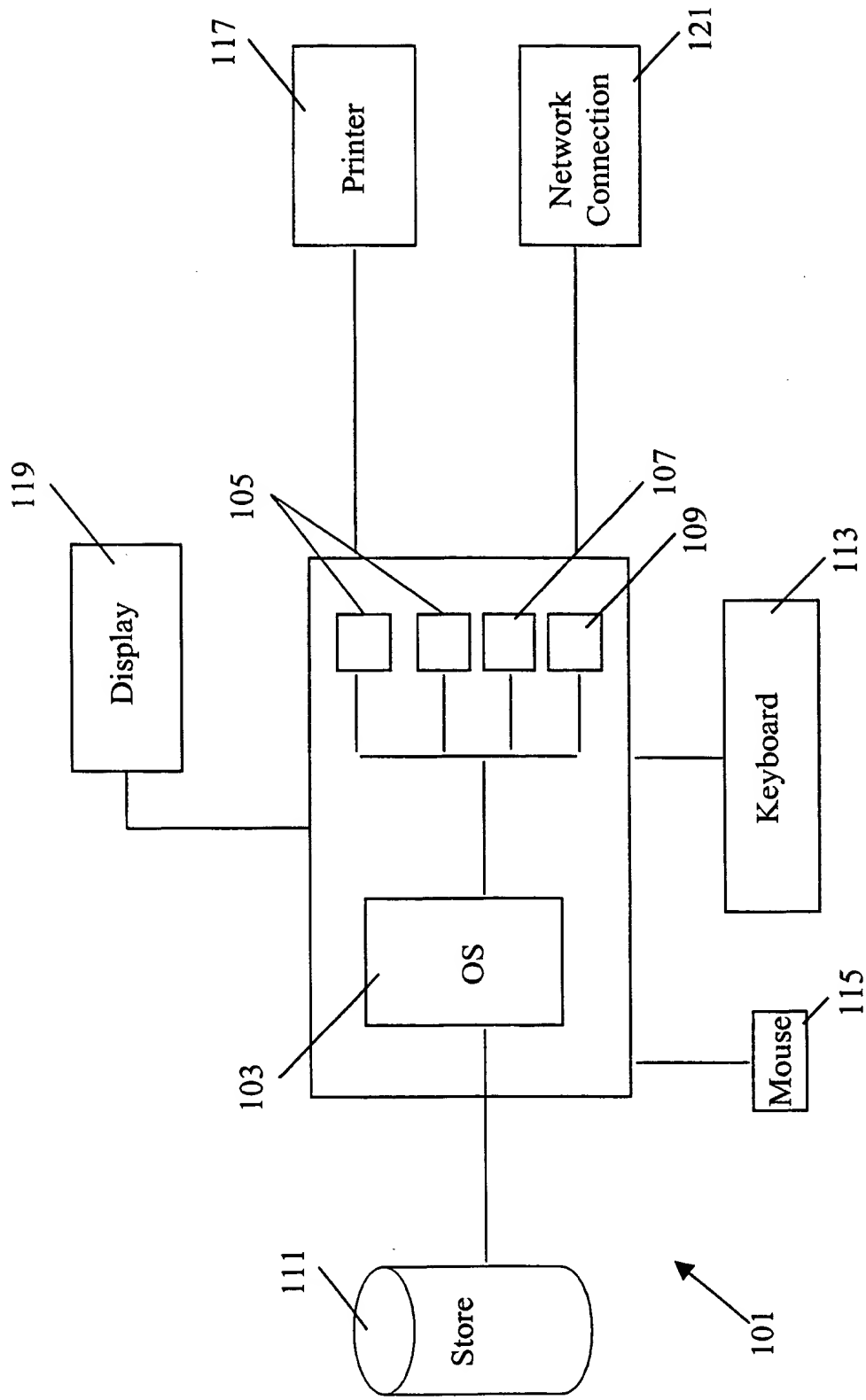

117
Printer
121
Network Connection
119
Display
105
107
109
113
Keyboard
OS
103
115
Mouse
111
Store
101

Figure 1

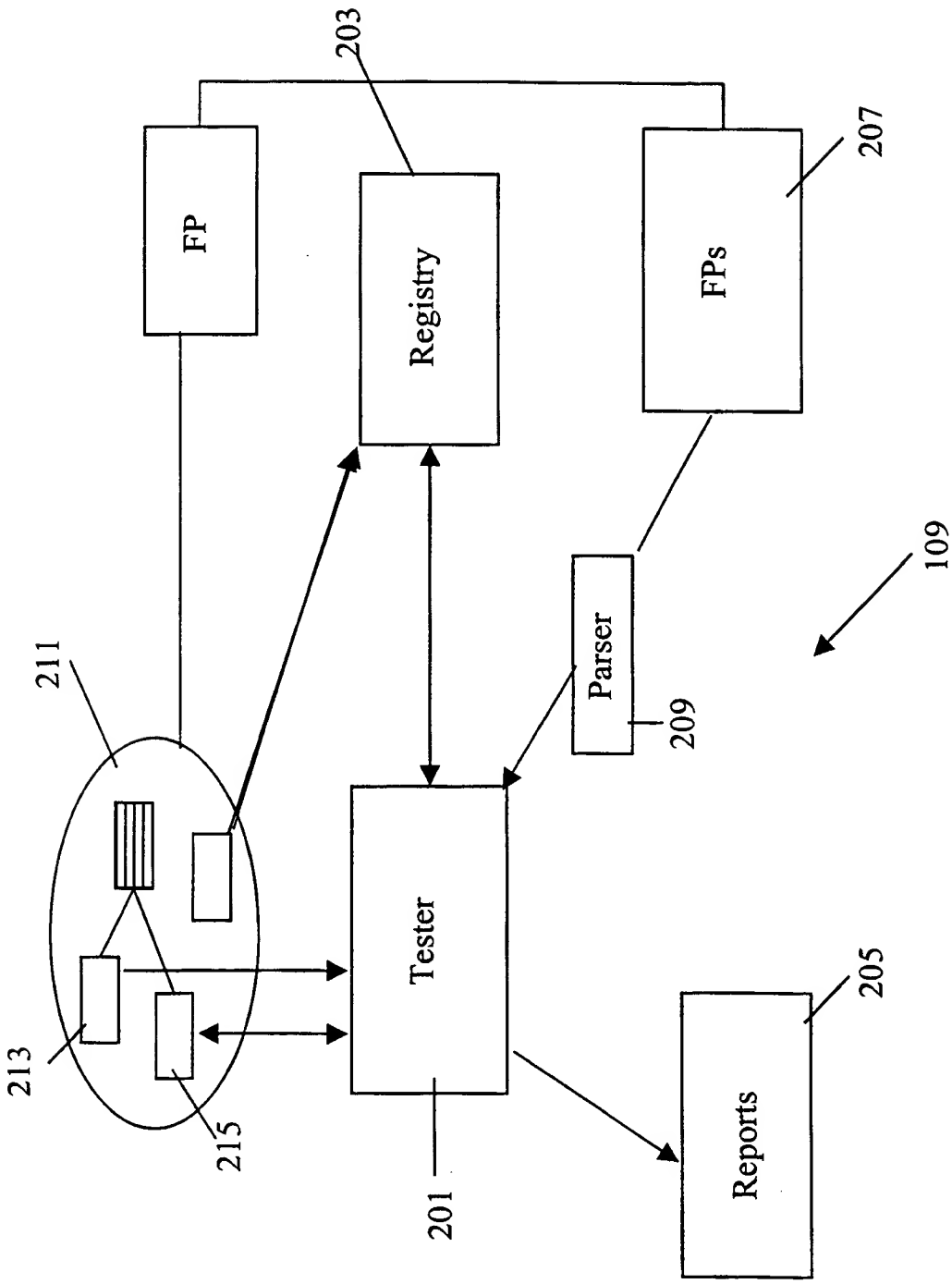

FP
203
Registry
FPs
207
211
Parser
209
109
213
215
Tester
201
Reports
205

Figure 2

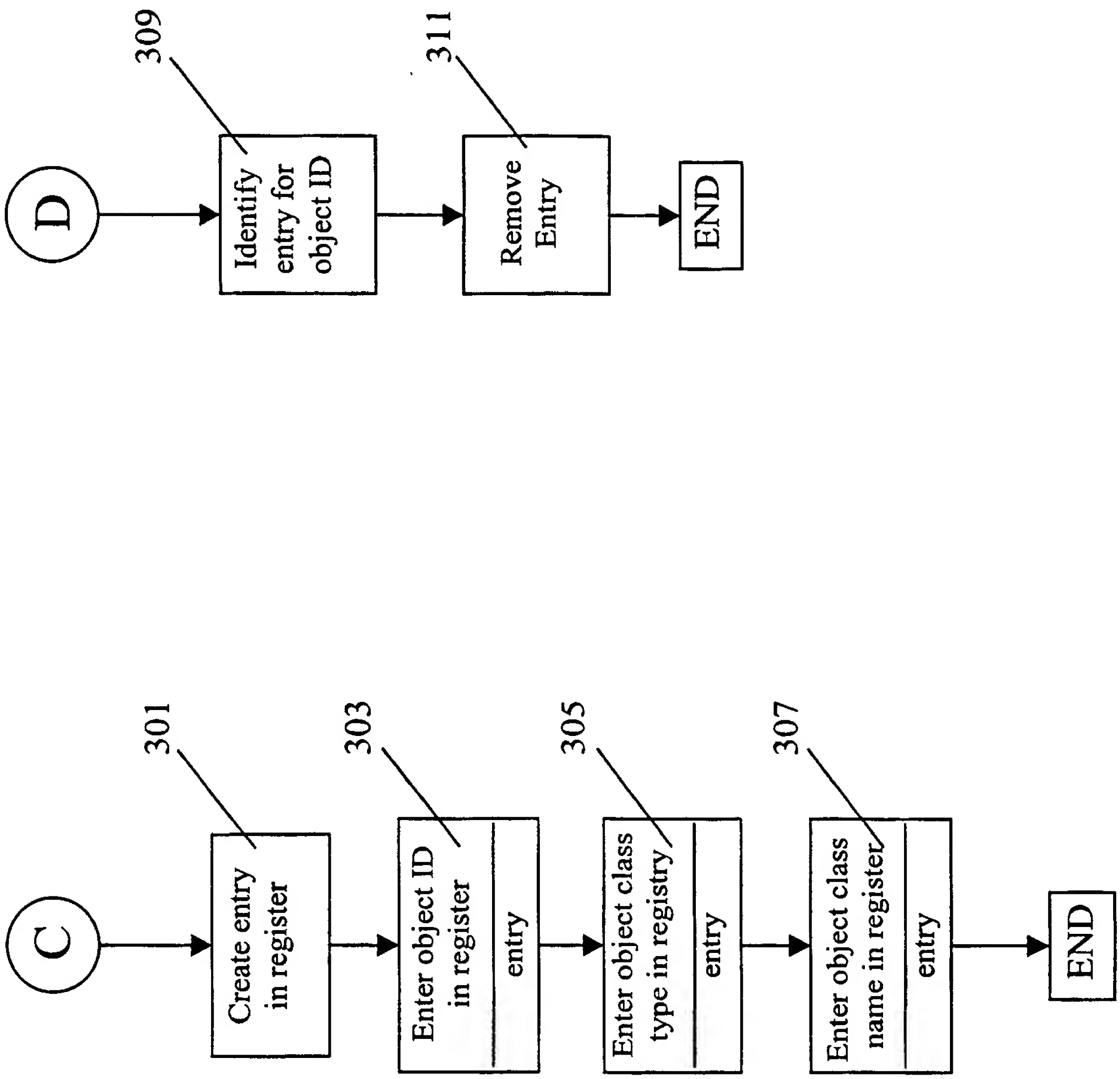

D
Identify entry for object ID
309
Remove Entry
311
END
C
Create entry in register
301
Enter object ID in register
entry
303
Enter object class type in registry
entry
305
Enter object class name in register
entry
307
END

Figure 3

Figure 4a

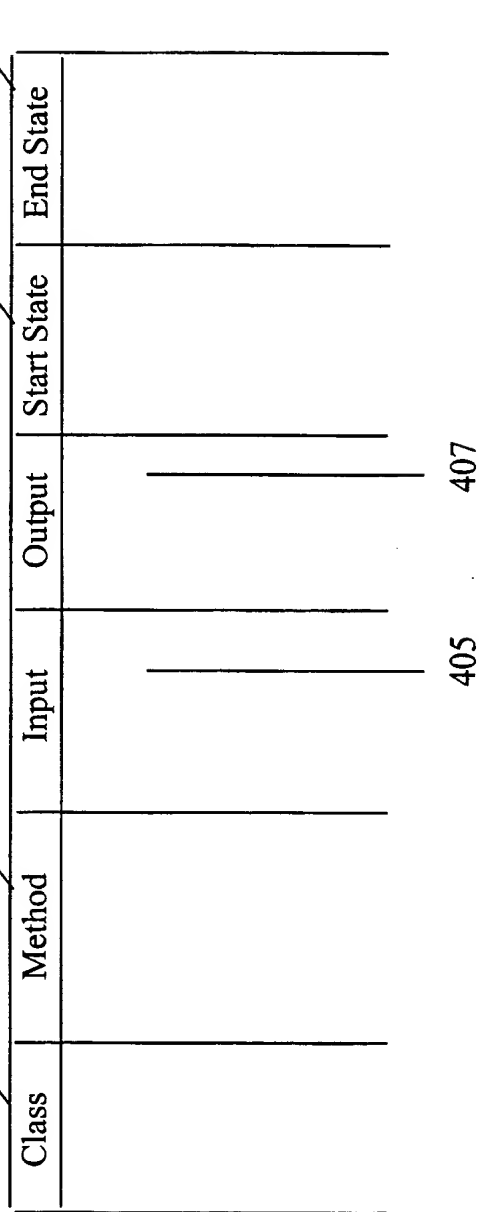


Figure 4b

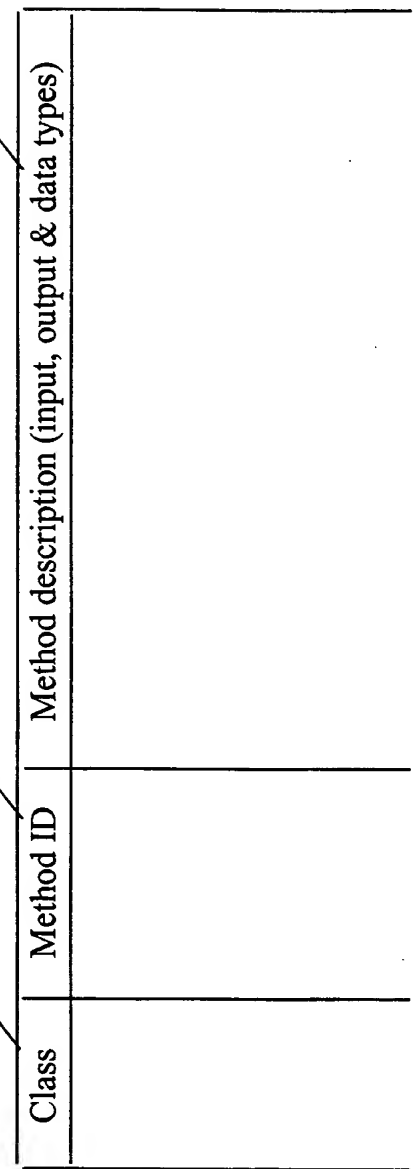
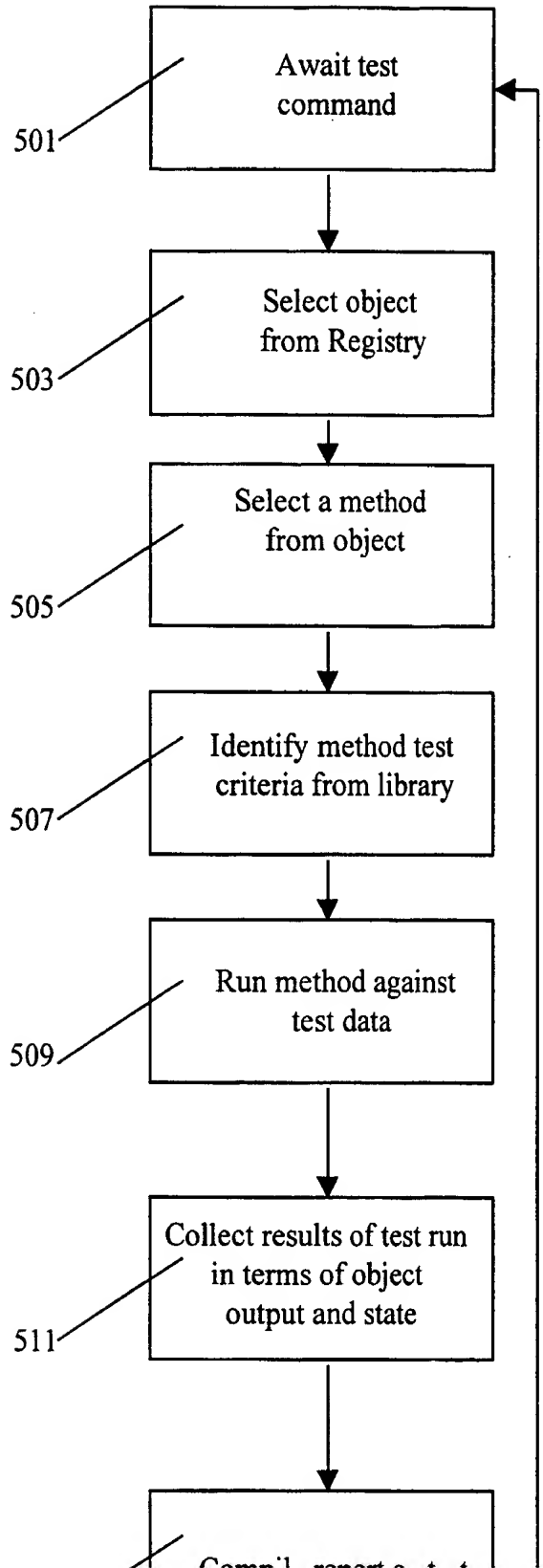


Figure 5


Await test command
501
Select object from Registry
503
Select a method from object
505
Identify method test criteria from library
507
Run method against test data
509
Collect results of test run in terms of object output and state
511

INTERNATIONAL SEARCH REPORT

Interi nal Application No
PCT/GB 00/00882

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G06F11/36

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	ROBERT V. BINDER: "Design for Testability in Object-Oriented Systems" COMMUNICATIONS OF THE ASSOCIATION FOR COMPUTING MACHINERY., vol. 37, no. 9, September 1994 (1994-09), pages 87-101, XP000485275 ASSOCIATION FOR COMPUTING MACHINERY. NEW YORK., US ISSN: 0001-0782 the whole document	1-10

☐ Further documents are listed in the continuation of box C.

☐ Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
22 June 2000	29/06/2000

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31–70) 340–2040, Tx. 31 651 epo nl,	Connemans, C

16

PATENT COOPERATION TREATY

PCT

INTERNATIONAL PRELIMINARY EXAMINATION REPORT

(PCT Article 36 and Rule 70)

Applicant's or agent's file reference	FOR FURTHER ACTION	See Notification of Transmittal of International Preliminary Examination Report (Form PCT/IPEA/416)
A25721 WO		
International application No.	International filing date *(day/month/year)*	Priority date *(day/month/year)*
PCT/GB00/00882	10/03/2000	31/03/1999
International Patent Classification (IPC) or national classification and IPC G06F11/36		
Applicant BRITISH TELECOMMUNICATIONS PUBLIC LIM. COMP.et al.		

1. This international preliminary examination report has been prepared by this International Preliminary Examining Authority and is transmitted to the applicant according to Article 36.

2. This REPORT consists of a total of 5 sheets, including this cover sheet.

   ☐ This report is also accompanied by ANNEXES, i.e. sheets of the description, claims and/or drawings which have been amended and are the basis for this report and/or sheets containing rectifications made before this Authority (see Rule 70.16 and Section 607 of the Administrative Instructions under the PCT).

   These annexes consist of a total of sheets.

3. This report contains indications relating to the following items:

   - I ☒ Basis of the report
   - II ☐ Priority
   - III ☐ Non-establishment of opinion with regard to novelty, inventive step and industrial applicability
   - IV ☐ Lack of unity of invention
   - V ☒ Reasoned statement under Article 35(2) with regard to novelty, inventive step or industrial applicability; citations and explanations suporting such statement
   - VI ☐ Certain documents cited
   - VII ☒ Certain defects in the international application
   - VIII ☐ Certain observations on the international application

Date of submission of the demand	Date of completion of this report
14/08/2000	11.12.2000
Name and mailing address of the international preliminary examining authority: European Patent Office D-80298 Munich	Authorized officer Bauer, R

INTERNATIONAL PRELIMINARY EXAMINATION REPORT

International application No. PCT/GB00/00882

I. Basis of the report

1. This report has been drawn on the basis of *(substitute sheets which have been furnished to the receiving Office in response to an invitation under Article 14 are referred to in this report as "originally filed" and are not annexed to the report since they do not contain amendments (Rules 70.16 and 70.17).)*:
Description, pages:

1-7 as originally filed

Claims, No.:

1-12 as originally filed

Drawings, sheets:

1-5 as originally filed

2. With regard to the **language**, all the elements marked above were available or furnished to this Authority in the language in which the international application was filed, unless otherwise indicated under this item.

These elements were available or furnished to this Authority in the following language: , which is:

- ☐ the language of a translation furnished for the purposes of the international search (under Rule 23.1(b)).
- ☐ the language of publication of the international application (under Rule 48.3(b)).
- ☐ the language of a translation furnished for the purposes of international preliminary examination (under Rule 55.2 and/or 55.3).

3. With regard to any **nucleotide and/or amino acid sequence** disclosed in the international application, the international preliminary examination was carried out on the basis of the sequence listing:

- ☐ contained in the international application in written form.
- ☐ filed together with the international application in computer readable form.
- ☐ furnished subsequently to this Authority in written form.
- ☐ furnished subsequently to this Authority in computer readable form.

☐ the drawings, sheets:

5. ☐ This report has been established as if (some of) the amendments had not been made, since they have been considered to go beyond the disclosure as filed (Rule 70.2(c)):

(Any replacement sheet containing such amendments must be referred to under item 1 and annexed to this report.)

6. Additional observations, if necessary:

V. Reasoned statement under Article 35(2) with regard to novelty, inventive step or industrial applicability; citations and explanations supporting such statement

1. Statement

Novelty (N)	Yes:	Claims	1-12
	No:	Claims	
Inventive step (IS)	Yes:	Claims	1-12
	No:	Claims	
Industrial applicability (IA)	Yes:	Claims	1-12
	No:	Claims	

2. Citations and explanations
see separate sheet

VII. Certain defects in the international application

The following defects in the form or contents of the international application have been noted:
see separate sheet

1 Reference is made to the following document:

D1: ROBERT V. BINDER: 'Design for Testability in Object-Oriented Systems' COMMUNICATIONS OF THE ASSOCIATION FOR COMPUTING MACHINERY., vol. 37, no. 9, September 1994 (1994-09), pages 87-101, XP000485275 ASSOCIATION FOR COMPUTING MACHINERY. NEW YORK., US ISSN: 0001-0782

ANNEX TO SECTION V

1 As explained below, independent **claim 1** appears to be new and inventive.

Claim 1 relates to a method of testing an operational integrated software system. Each active element of software is automatically registered and associated with a set of test criteria. An element of software is selected from the registered elements and tested in accordance with the associated set of test criteria. The results of the testing are captured and compared to the associated test criteria.

The ISR cites only the document D1. D1 is, therefore, considered the closest prior art document. It describes a method of testing an object-oriented system. In D1 a test driver generates a set of test criteria from a test specification belonging to an element of software. The test driver tests the element of software in accordance with the generated test criteria and captures and evaluates the results. However, D1 merely describes how a single element of software is tested. It is neither described nor suggested in D1 which or how elements of software are selected from the object oriented system for testing. In contrast thereto, **claim 1** defines a method for testing all active elements of software of the operational integrated software system by *automatically registering each active element of software* and *selecting from these registered elements an element of software for test.*

Therefore, the subject-matter of **claim 1** is new and inventive and meets the requirements of *Article 33(2) and 33(3) PCT*

2 **Claims 2-5** are dependent on **claim 1** and add further features to the inventive method of **claim 1**. Therefore, the subject-matters of **claims 2-5** are new and inventive and meet the requirements of *Article 33(2) and 33(3) PCT*.

3 Each of the **system claims 6-10** has a counterpart in the method **claims 1-5**. Therefore, the same observations as for method **claims 1-5** apply for system **claims 6-10**. Thus, the subject-matters of **claims 6-10** are new and inventive and meet the requirements of *Article 33(2) and 33(3) PCT*.

4 **Claim 11** defines a data carrier carrying instructions for causing the computer to carry out the method according to **claim 1**. **Claim 12** defines a data carrier carrying instructions for enabling the computer to provide the system according to **claim 6.** Therefore, the same observations as for **claims 1 and 6** apply for **claims 11 and 12.** Thus, the subject-matters of **claims 11 and 12** are new and inventive and meet the requirements of *Article 33(2) and 33(3) PCT*.

ANNEX to SECTION VII

1 To meet the requirements of *Rule 5.1(a)(ii) PCT*, the closest prior art document D1 should have been identified in the description and the relevant background art disclosed therein should have been briefly discussed.

2 The independent claims should have been drafted in the two-part form in accordance with *Rule 6.3(b) PCT* with those features known from prior art document D1 being placed in the preamble and the remaining features being included in the characterizing part.

3 The features of the system claims should have been provided with reference signs